SPYWOLF

Security Audit Report

(TESTNET)

Completed on
January 31, 2023

SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

OVERVIEW

This audit has been prepared for **North Apes** to review the main aspects of the project to help investors make make an informative decision during their research process.

You will find a a summarized review of the following key points:

- ✓ Contract's source code
- ✓ Owners' wallets
- ✓ Tokenomics
- ✓ Team transparency and goals
- ✓ Website's age, code, security and UX
- ✓ Whitepaper and roadmap
- ✓ Social media & online presence

> *The results of this audit are purely based on the team's evaluation and does not guarantee nor reflect the projects outcome and goal*
>
> - SPYWOLF Team -

TABLE OF CONTENTS

North Apes

PROJECT DESCRIPTION

According to their whitepaper:

Release Date: Presale starts in February, 2023

Category:

CONTRACT INFO

Token Name North COin	Symbol NORTH
Contract Address 0x84693a1FFc25ea7e03d9783Ab04276B4EA598eB8	
Network Binance Smart Chain TESTNET	Language Solidity
Deployment Date Jan 30, 2023	Verified? Yes
Total Supply 1,000,000,000	Status Not launched

TAXES

Buy Tax
Up to 5%

Sell Tax
Up to 25%

Our Contract Review Process

The contract review process pays special attention to the following:

- ✓ Testing the smart contracts against both common and uncommon vulnerabilities
- ✓ Assessing the codebase to ensure compliance with current best practices and industry standards.
- ✓ Ensuring contract logic meets the specifications and intentions of the client.
- ✓ Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- ✓ Thorough line-by-line manual review of the entire codebase by industry experts.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

CURRENT STATS

(As of January 31, 2023)

Status:

Not Launched!

Liquidity

Not added yet

Burn

No burnt tokens

MaxTxAmount
No limit

DEX
PancakeSwap

LP Address(es)

Liquidity not added yet

TOKEN TRANSFERS STATS

Transfer Count	TESTNET
Uniq Senders	TESTNET
Uniq Receivers	TESTNET
Total Amount	TESTNET
Median Transfer Amount	TESTNET
Average Transfer Amount	TESTNET
First transfer date	TESTNET
Last transfer date	TESTNET
Days token transferred	TESTNET

SMART CONTRACT STATS

Calls Count	TESTNET
External calls	TESTNET
Internal calls	TESTNET
Transactions count	TESTNET
Uniq Callers	TESTNET
Days contract called	TESTNET
Last transaction time	TESTNET
Created	TESTNET
Create TX	TESTNET
Creator	TESTNET

FEATURED WALLETS

Owner address	0xdb5dc1e4c1841537248545f1d3ad9f5679c2c13f
LP address	**Liquidity not added yet**

TOP 3 UNLOCKED WALLETS

1 N/A

2 N/A

3 N/A

VULNERABILITY CHECK

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions and reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

THREAT LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

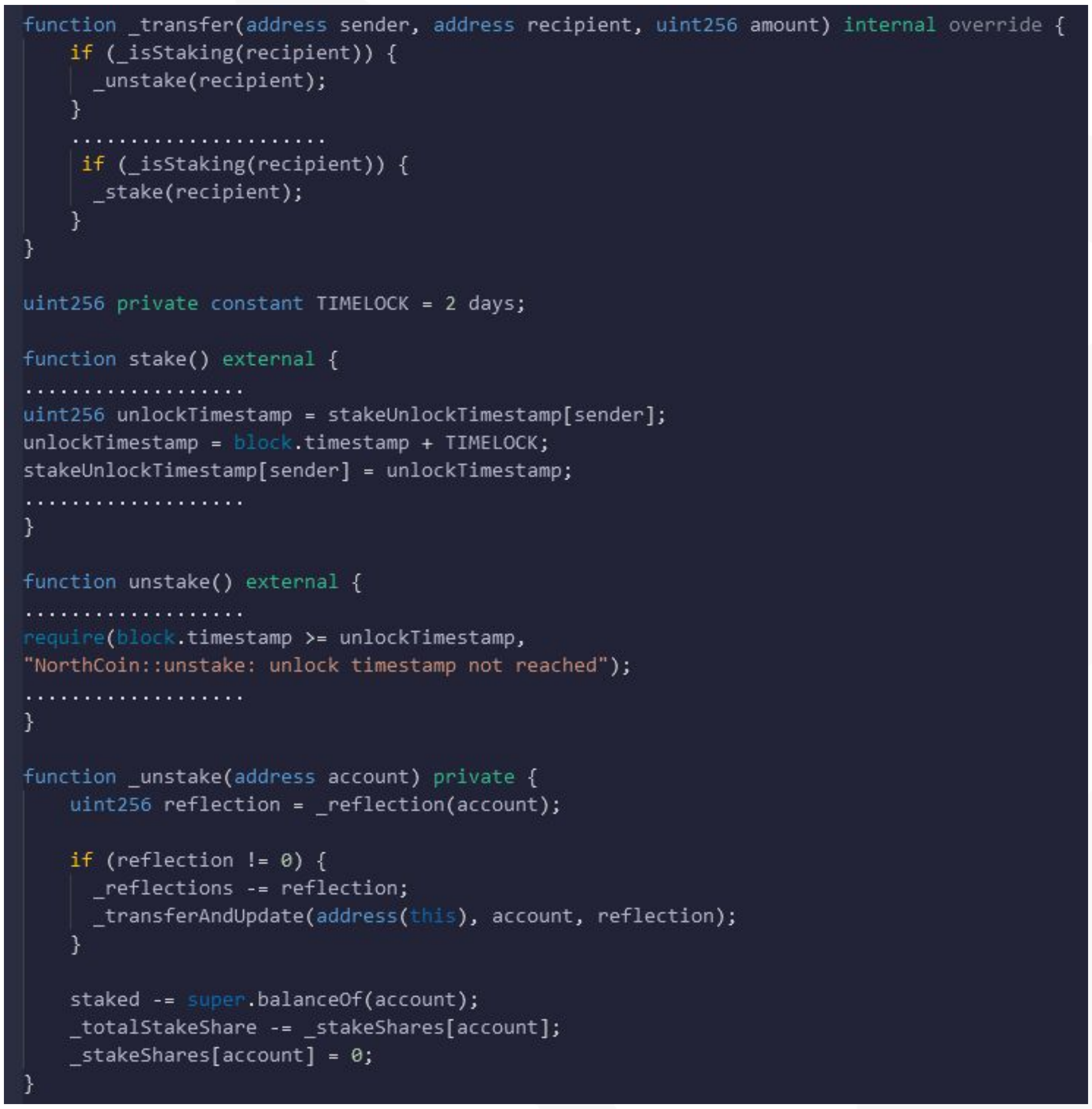
Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

FOUND THREATS

High Risk

Reflections from staking pool can be drained from fraudulent user.



```
function _transfer(address sender, address recipient, uint256 amount) internal override {
    if (_isStaking(recipient)) {
      _unstake(recipient);
    }
    ......................
     if (_isStaking(recipient)) {
      _stake(recipient);
    }
}

uint256 private constant TIMELOCK = 2 days;

function stake() external {
...................
uint256 unlockTimestamp = stakeUnlockTimestamp[sender];
unlockTimestamp = block.timestamp + TIMELOCK;
stakeUnlockTimestamp[sender] = unlockTimestamp;
...................
}

function unstake() external {
...................
require(block.timestamp >= unlockTimestamp,
"NorthCoin::unstake: unlock timestamp not reached");
...................
}

function _unstake(address account) private {
    uint256 reflection = _reflection(account);

    if (reflection != 0) {
      _reflections -= reflection;
      _transferAndUpdate(address(this), account, reflection);
    }

    staked -= super.balanceOf(account);
    _totalStakeShare -= _stakeShares[account];
    _stakeShares[account] = 0;
}
```

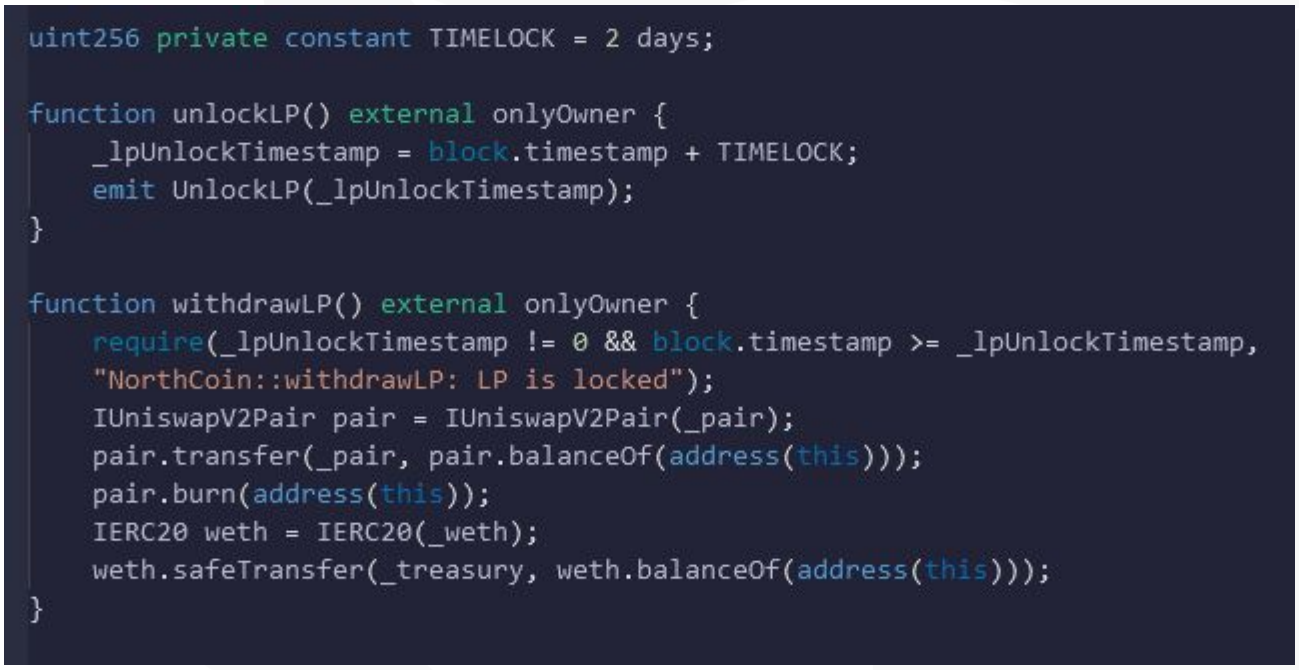
POC:
User A, B, C, D stake significant amount of tokens. Meanwhile staking pool grows from taxes.
User E stake some amount of tokens.
User E deploy contract with which he sends multiple small amounts of tokens to himself (staked account) in 1 tx or do this manually many times.
Result - The entire staking pool is drained by user E, bypassing the TIMELOCK variable.

- Recommendation:
    - Consider another transfer mechanism to staked users.

FOUND THREATS

⚠ Medium Risk

Owner can withdraw LP tokens from the contract 2 days after initiating the unlockLP function.
Using the function on testnet contracts failed with 'INSUFFICIENT_LIQUIDITY_BURNED' error from the pair's contract burn function.



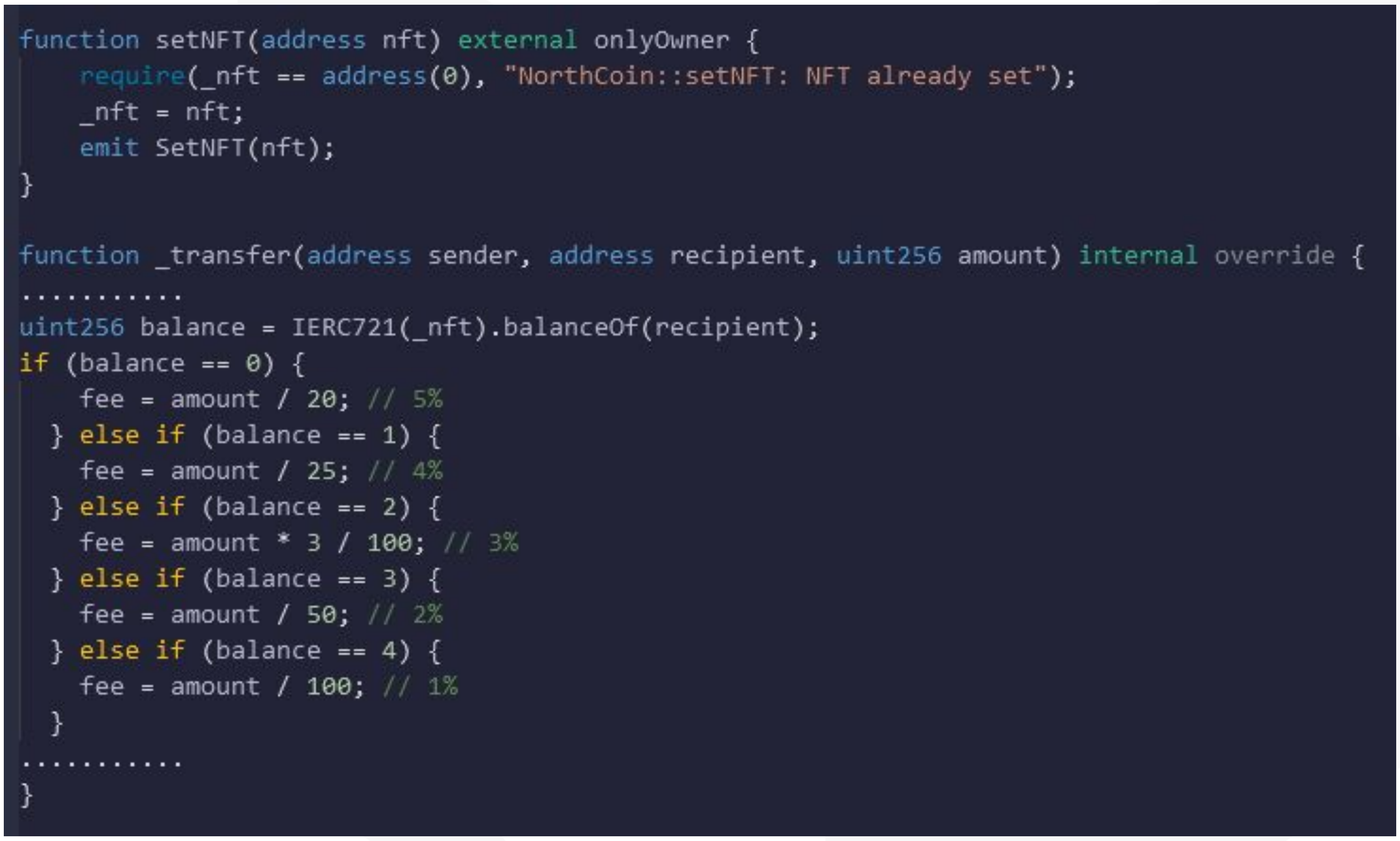
```
uint256 private constant TIMELOCK = 2 days;

function unlockLP() external onlyOwner {
    _lpUnlockTimestamp = block.timestamp + TIMELOCK;
    emit UnlockLP(_lpUnlockTimestamp);
}

function withdrawLP() external onlyOwner {
    require(_lpUnlockTimestamp != 0 && block.timestamp >= _lpUnlockTimestamp,
    "NorthCoin::withdrawLP: LP is locked");
    IUniswapV2Pair pair = IUniswapV2Pair(_pair);
    pair.transfer(_pair, pair.balanceOf(address(this)));
    pair.burn(address(this));
    IERC20 weth = IERC20(_weth);
    weth.safeTransfer(_treasury, weth.balanceOf(address(this)));
}
```

- Recommendation:
    - Simpler methods to clear tokens from contract can be more efficient and less prone to errors.
      Sample: IERC20(token).transfer(to, amount)

FOUND THREATS

⚠ Medium Risk

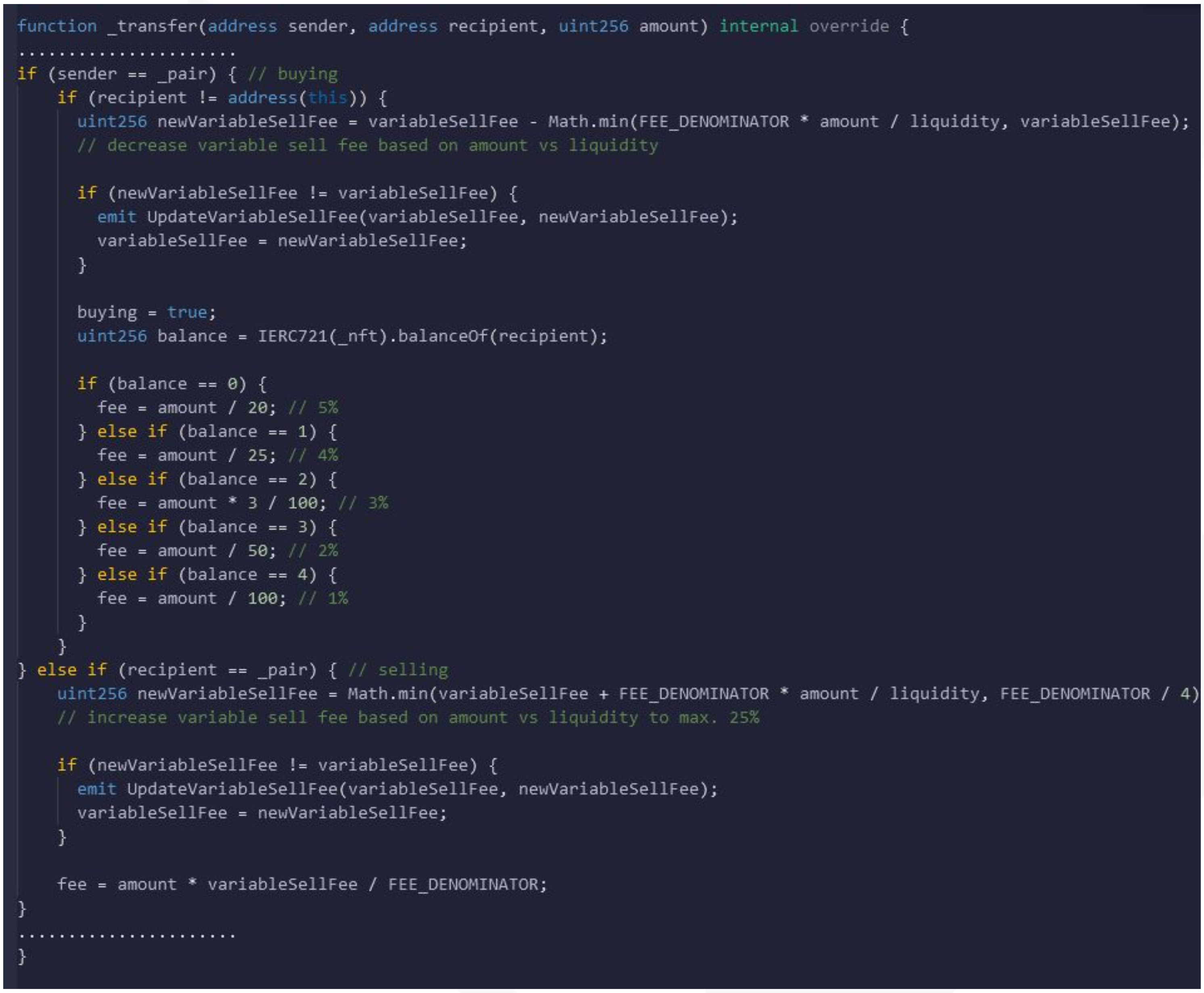
Owner can change the NFT address.
If non contract address is set as NFT or contract that do not have the correct balanceOf() function, trading will fail.



```
function setNFT(address nft) external onlyOwner {
    require(_nft == address(0), "NorthCoin::setNFT: NFT already set");
    _nft = nft;
    emit SetNFT(nft);
}

function _transfer(address sender, address recipient, uint256 amount) internal override {
...........
uint256 balance = IERC721(_nft).balanceOf(recipient);
if (balance == 0) {
    fee = amount / 20; // 5%
  } else if (balance == 1) {
    fee = amount / 25; // 4%
  } else if (balance == 2) {
    fee = amount * 3 / 100; // 3%
  } else if (balance == 3) {
    fee = amount / 50; // 2%
  } else if (balance == 4) {
    fee = amount / 100; // 1%
  }
...........
}
```

⚠ Low Risk

This token uses dynamic sell fees up to 25%.
Buy fees are from 5% to 1% considering the amount of NFTs that user hold (from 0 to 4 NFTS).
Combined buy+sell=30%.

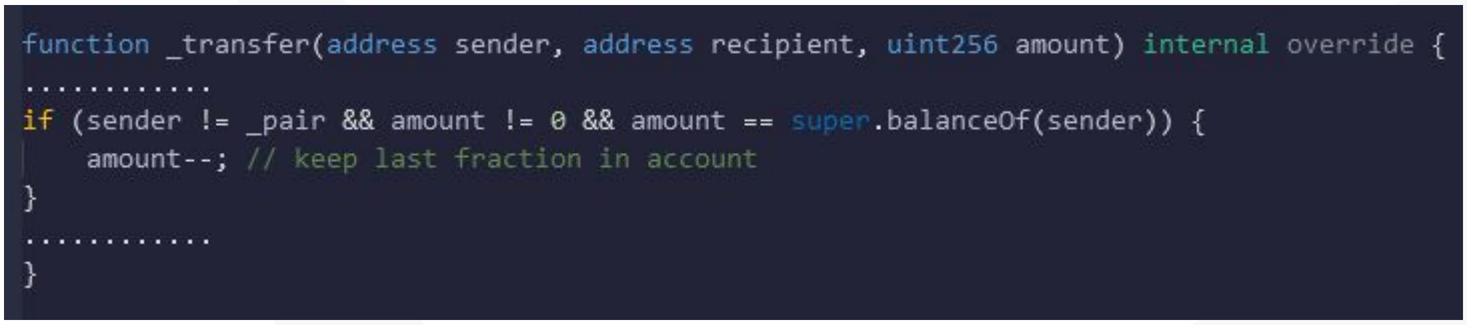


```
function _transfer(address sender, address recipient, uint256 amount) internal override {
......................
if (sender == _pair) { // buying
    if (recipient != address(this)) {
      uint256 newVariableSellFee = variableSellFee - Math.min(FEE_DENOMINATOR * amount / liquidity, variableSellFee);
      // decrease variable sell fee based on amount vs liquidity

      if (newVariableSellFee != variableSellFee) {
        emit UpdateVariableSellFee(variableSellFee, newVariableSellFee);
        variableSellFee = newVariableSellFee;
      }

      buying = true;
      uint256 balance = IERC721(_nft).balanceOf(recipient);

      if (balance == 0) {
        fee = amount / 20; // 5%
      } else if (balance == 1) {
        fee = amount / 25; // 4%
      } else if (balance == 2) {
        fee = amount * 3 / 100; // 3%
      } else if (balance == 3) {
        fee = amount / 50; // 2%
      } else if (balance == 4) {
        fee = amount / 100; // 1%
      }
    }
} else if (recipient == _pair) { // selling
    uint256 newVariableSellFee = Math.min(variableSellFee + FEE_DENOMINATOR * amount / liquidity, FEE_DENOMINATOR / 4)
    // increase variable sell fee based on amount vs liquidity to max. 25%

    if (newVariableSellFee != variableSellFee) {
      emit UpdateVariableSellFee(variableSellFee, newVariableSellFee);
      variableSellFee = newVariableSellFee;
    }

    fee = amount * variableSellFee / FEE_DENOMINATOR;
}
......................
}
```

- Recommendation:
    - Considered as good tax deduction practice is buy and sell fees combined not to exceed 25%.

⚠ Low Risk

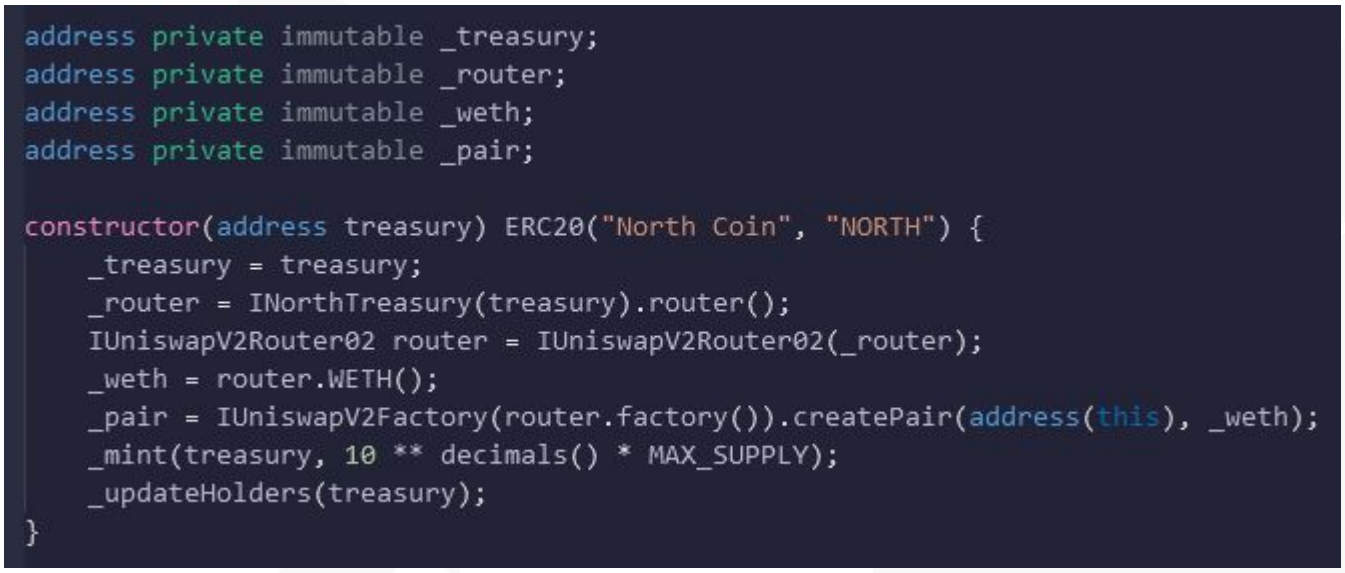
Once bought, user will always have fraction of a token which will keep holders keep growing constantly.
This will show biased stats for holders, because user will be considered as holder even if they have amounts small such as 0.0000000000000000000000001 token.



```
function _transfer(address sender, address recipient, uint256 amount) internal override {
............
if (sender != _pair && amount != 0 && amount == super.balanceOf(sender)) {
    amount--; // keep last fraction in account
}
............
}
```

Informational

This contract uses intermediary 'treasury' contract that sets the token's router and receives the total minted token supply and NFT contract that is used in dynamic buy taxes.
The intermediary contract and the NFT contract are not in the scope of the current audit.



```
address private immutable _treasury;
address private immutable _router;
address private immutable _weth;
address private immutable _pair;

constructor(address treasury) ERC20("North Coin", "NORTH") {
    _treasury = treasury;
    _router = INorthTreasury(treasury).router();
    IUniswapV2Router02 router = IUniswapV2Router02(_router);
    _weth = router.WETH();
    _pair = IUniswapV2Factory(router.factory()).createPair(address(this), _weth);
    _mint(treasury, 10 ** decimals() * MAX_SUPPLY);
    _updateHolders(treasury);
}
```

RECOMMENDATIONS FOR

GOOD PRACTICES

1 Consider fundamental tradeoffs

2 Be attentive to blockchain properties

3 Ensure careful rollouts

4 Keep contracts simple

5 Stay up to date and track development

North Apes

GOOD PRACTICES FOUND

✓ The owner cannot mint new tokens after deployment

✓ The owner cannot set a transaction limit

✓ The smart contract utilizes "Math" to prevent overflows

SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- ✓ OVER 150 SUCCESSFUL CLIENTS
- ✓ MORE THAN 500 SCAMS EXPOSED
- ✓ MILLIONS SAVED IN POTENTIAL FRAUD
- ✓ PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS
- ✓ CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

FIND US ONLINE

SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

SPYWOLF.NETWORK

@SPYWOLFOFFICIAL

@SPYWOLFNETWORK

Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.

While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER:

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.

No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.

The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.

No applications were reviewed for security. No product code has been reviewed.